An R Tutorial for the Non-Coding-Inclined

As empirical, quantitative methods continue to become more commonplace in linguistic research, linguists are increasingly employing various kinds of experimental tasks, such as grammaticality judgment tasks, word/phrase-list readings, perceptual discrimination tasks, matched guise techniques, sociolinguistic interviews, surveys/questionnaires, pre-/post-tests, and corpus analyses. In order to be able to analyze the distinct kinds of data obtained from these and other tasks, minimally, a passive knowledge of inferential statistics and a degree of proficiency with inferential statistics software are required.

Two principal complications present themselves with respect to these requirements: (1) the set of statistical techniques deemed appropriate for specific types of data is inherently non-static, evolving as advances are made in the statistics discipline, effectively resulting in ‘waves’ of consensus in the Linguistics community regarding the use of particular statistical tests for linguistic data, and; (2) multiple software options exist and continue to be developed for performing inferential statistics, each with unique interfaces and capabilities that evolve alongside the very tests they were designed to perform. Thus, a linguist’s endeavor to gain expertise in experimental methodologies, statistical theory, and statistical software packages is much more akin to chasing a perpetually moving target than mastering a finite set of skills.

One software package in particular, R ([3]), has arguably become the new norm for statistical analysis in the Linguistics community, boasting a free and maximally powerful open-source platform that stands in stark contrast to other competitors (e.g. SAS [4], SPSS [1], STATA [6]), that are only accessible through paid (and often University) subscription. Unfortunately, however, R’s minimal user interface begs programming language, constituting a daunting and perhaps unintuitive burden for many linguists. Still, as efforts to combat this reliance on user-generated code remain somewhat restricted to Variationist Sociolinguistics (e.g. Rbrul [2], Language Variation Suite [5]), it seems increasingly likely that the newer generations of empirical linguists will be tasked with becoming proficient in R.

My goal is to all but eliminate R’s barrier to entry, affording fuller access to statistical analyses to as wide a community of linguists as possible. Reducing user-generated coding to the absolute minimum, namely the typing out of names of independent and dependent variables of interest, this Copy-Paste guide helps enable users to fully perform R analyses with linguistic data. Accompanying files (a version of this guide as an R file to make copy-pasting easier [i.e., no need to click back and forth between R and your PDF-viewer], as well as example datasets and example data organization templates) are available at the bottom of:

https://spanish-portuguese.berkeley.edu/people/justin-davidson/

Please note: This is NOT a guide for or declaration of ‘correct’ statistical techniques, which should rightly come from the field of statistics, rather than linguistics. R will perform the commands you give it, even if those commands are statistically unsound!

References


TABLE OF CONTENTS

The Only Coding Knowledge You’ll Need ................................................................. Page 3
Tips and Terminology .............................................................................................. Page 4
Which Test Do I Use??? .......................................................................................... Page 5
Understanding an ANOVA output vs. a Regression output ..................................... Page 6
Reporting Linear Regression and ANOVA ............................................................. Page 8
Reporting Logistic Regression ................................................................................ Page 9

Between-Subjects ANOVA ......................................................................................... Page 10
Within-Subjects ANOVA ......................................................................................... Page 11
Fixed Effects Linear Regression ............................................................................... Page 13
Fixed Effects Logistic Regression ............................................................................ Page 15
Mixed Effects Linear Regression ............................................................................. Page 17
Mixed Effects Logistic Regression ........................................................................... Page 19
Poisson & Zero-Inflated Poisson Regression .......................................................... Page 21
Model Comparison ................................................................................................ Page 23
Step-Wise Regression .............................................................................................. Page 24
Chi-Square Test ....................................................................................................... Page 25

Unfortunate Problems............................................................................................. Page 26
The ONLY Coding Knowledge You’ll Need
(A mini-version of this appears at the top of every test, just as a helpful reminder)

The following colored abbreviations are the only code you will need to create/write:

V - Any variable (independent or dependent), i.e., the name of one of your Excel/spreadsheet columns
IV - An independent variable, i.e., the name of one of your Excel/spreadsheet columns
DV - A dependent variable, i.e., the name of one of your Excel/spreadsheet columns
# - Input a number (such as 2 or 3). Specifics on what kind of number to insert appear beside codes that require this.

IVDUMP - Your syntax for fixed main effects and interactions. IVs are separated by a plus sign and interactions are denoted by an asterisk. Some examples appear below, for a hypothetical experiment with the following 3 IVs: Gender, Age, and Verb

Gender + Age Tests for main effects of gender and age, one at a time.
Gender * Age Tests for main effects of gender and age, one at a time, and also tests for their interaction (i.e., does a gender effect depend on age?. Does an age effect depend on gender?)
Gender + Age * Verb Tests for main effects of each IV, 1 at a time, and also tests for 1 interaction (age/verb).
Verb * Age + Gender No different from the code immediately above this. You can transpose items freely; it's the signs (+ or *) between them that matter.
Gender * Age + Verb * Age Tests for main effects of each IV, 1 at a time, as well as 2 interactions (age/gender & age/verb)
Gender * Age + Gender The "+ Gender" is redundant, since Gender * Age already tests for each IV separately. Delete "+ Gender" and reduce to simply "Gender * Age".

RIVDUMP - Your syntax for random intercepts/effects and slopes in a mixed effects model. Random effects/intercepts and slopes appear in enclosed parentheses, such as (1|FirstIntercept) + (1|SecondIntercept), each one separated by a plus sign. Use the "1" when no slope is present, and replace the "1" with a within-subjects IV to include it as a slope for the intercept in question. Examples appear below for an experiment with IVs of Gender, Verb, Word, and Participant:

Gender * Verb + (1|Participant) Fixed main and interaction effects for Gender/Verb, with Participant as a random intercept/effect.
Gender + Verb + (1|Participant) + (1|Word) Fixed main effects of Gender and Verb, with each of Participant and Word as random intercepts/effects.
Gender + Verb + (Verb|Participant) Fixed main effects of Gender and Verb, with Participant as a random intercept/effect that has Verb as a random slope.

ROWDUMP – Only applicable when forcing an ANOVA output for a fixed effects logistic regression. See explanation at the top of the fixed effects logistic regression page if you ever want to do this.
Tips and Terminology

- **ColoredWords** like **this** or **this** indicate labels that you can choose to rename, keeping labels consistent with colors across a given test. Personally, I recommend not modifying/editing them, since the names I’ve given them are consistent (and it’s one less thing you have to worry about [or potentially mess up]!).

- For each analysis you run, quit R and open it up anew. Otherwise, you'll find that it remembers objects like **DataName** and the like from past sessions, which will be problematic. (I.e., if you ever get an error about a **Name** being ‘masked’ from prior analyses, this means R remembers this object from a prior session and you need to quit R and start fresh. Clearing your workspace [clickable under the Workspace menu at the top of the screen] once you open up R again should do the trick, especially for RStudio users.)

- RStudio is an alternative software package to R for those that want a more snazzy view of things. Beyond seeing this cheat sheet (assuming you go look at my version of this tutorial as an R file) in the same window as your command-line window, RStudio also can be set to auto-load-in library packages (i.e., every step 2 for each test), which can let you skip some copy-pasting and begin each analysis at step 3.

- Every test begins with a few command lines of **purple italics**, namely **install.packages("SOMETHING")**. You need to be connected to the Internet to successfully run these lines. Luckily, after you run them once, you do NOT have to run them again unless you re-download R (i.e., you update it, or alternatively you trash R and later decide to come crawling back ;) Accordingly, plan on usually skipping to step 2 (and never needing an Internet connection to run your analyses!).

- Tukey post-hoc tests via **emmeans** generate a warning note in red about misleading results. Don’t be alarmed when you see it!

**Nominal Variable**: A variable whose levels are discrete, non-numerical categories or labels, such as "Young" vs. "Old" for Age, or "Absent" vs. "Present" for Copula Use. Since these are discrete categories, you cannot take a meaningful average/mean of them, i.e., it doesn't make sense to ask "What's the average of "Young" and "Old"?" IMPORTANT – In your Excel/spreadsheet file, all nominal variables must have cells that begin with a letter (rather than a number). If you have an IV of Verb, with two levels: Type 1 and Type 2, then be sure your cells are **Type1** and **Type2** (or even **T1** and **T2**), rather than **1** and **2**.

**Continuous Variable**: A variable whose levels are numerical values that can be averaged, such as Formant Frequency (250 vs. 252 vs. 300 etc...) and Age (33 vs. 35 vs. 45 etc...). Note that a single variable can be treated as either nominal or continuous by the researcher, such as Age in the examples here (Continuous Age: 33 vs. 45, etc.; Nominal Age: Young vs. Old [where Young is anybody 18-30 and Old is anybody 45-60]). Be careful with nominal IVs that might be associated with numbers, such as Participant Number: Person 1, Person 2, Person 3, etc. You wouldn't want to say that there exists an average participant number of 2 (via [1+2+3]/3) in an experiment with Person 1, Person 2, and Person 3. IMPORTANT – In your Excel/spreadsheet file, all continuous variables must have cells that are numerical (i.e., no letters!). For the example of Participant Number, which should be a nominal IV, do NOT fill cells as **1**, **2**, and **3**, and instead have them start with a letter (e.g. **p1**, **p2**, **p3**).

**Between-subjects IV** = a nominal IV for which each participant can only contribute a DV response for 1 level. Social IVs (gender, country of origin, age group, language dominance group, etc.) are commonly between-subjects IVs, since a participant is static during an experiment (i.e., a participant can't be classified as "From Argentina" for some of the experiment and "From Uruguay" for another portion of it. Each participant represents or contributes DV data for only 1 level of each between-subjects IV.)

**Within-subjects IV** = a nominal IV for which each participant contributes a DV response for all/each of the IV's levels. These are often linguistic IVs, such as Phrase Type, whereby every participant offers a DV response for every type of phrase.
Which Test Do I Use???  *(Flowchart-esque instructions limited to tests covered here only)*

**Question 1: What kind of Dependent Variable do you have?**

**Possible answer 1: Continuous**
- Between-subjects ANOVA (if all IVs are nominal and between-subjects, with NO random effects. **Question 2** does not apply.)
- Within-subjects ANOVA (if all IVs are nominal and 1+ is within-subjects, NO random effects. **Question 2** does not apply.)
- Some type of Linear Regression (no restrictions! Math is identical to between-/within-subjects ANOVAs, so you’re covered no matter what!)

**Possible answer 2: Nominal, Single-level (i.e., counted occurrences of a single, non-varying type of nominal response)**
- Chi-Square Test (if all IVs are nominal, with NO random effects. This test prevents you from running any IV interactions, and moreover can only be run 1 IV at a time, making it more problematic if you have several IVs. **Question 2** does not apply.)
- Some type of Poisson Regression (no restrictions! However, choice between Poisson vs. Zero-Inflated Poisson is discussed in the Poisson coding section, with the empirical test to motivate one vs. the other appearing as code step [3a] in Model Comparison.)

**Possible answer 3: Nominal, two-level (henceforth “binary”)**
- Some type of Logistic Regression (no restrictions!)

**Question 2: Do you have any random effects (i.e., random intercepts or random intercepts and slopes)?**

**Possible answer 1: No!**
- Fixed effects model (i.e., whichever regression you’re going to run, choose the FIXED effects version of it.)

**Possible answer 2: Yes!**
- Mixed effects model (i.e., whichever regression you’re going to run, choose the MIXED effects version of it.)

**Possible answer 3: I don’t know... should I?**
- Model Comparison (i.e., head to the Model Comparison section and follow the instructions to run code step [3b] on two models that are identical beyond one being FIXED [lacking random effects] and one being MIXED [having 1+ random effect].)

**Question 3: Do you already know the IVs you’re including and which ones are or are not tested as interaction terms?**

**Possible answer 1: Yes!**
- Peachy! Go run your stats!

**Possible answer 2: No!  ... or I thought I did, but now your question is making me second-guess myself!**
- Model Comparison (i.e., create 2 [or more] models and then compare them to decide which to run stats as normal on.)

**Possible answer 3: I'm a Variationist Sociolinguist that wants to do Variable Rule Analysis (Varbrul) a la Goldvarb!**

- Step-wise Regression (i.e., run a step-wise regression to identify the best-fit model and then run stats as normal on it. Note that the conversion formula for β-coefficients into factor weights appears in the codes for Logistic models, and a full example of the math appears on the Reporting Logistic Regression page.)
Understanding an ANOVA output vs. a Regression output

RQ: Do speakers of different ages and languages produce /a/ (F1) differently (i.e., does age and/or language affect /a/ production)?

Independent Variable #1 - Age: Young vs. Middle vs. Old
Independent Variable #2- Language: Spanish vs. Catalan
Dependent Variable: F1 in hertz

Hypothetical results of a production experiment:
- Young Spanish: 800 hz
- Young Catalan: 400 hz
- Middle Spanish: 400 hz
- Middle Catalan: 800 hz
- Old Spanish: 400 hz
- Old Catalan: 400 hz

An ANOVA output is expressed in terms of individual independent variables being statistically significant or not (i.e., do they or do they not affect the dependent variable). An ANOVA output for the aforementioned results might look like the following:

AGE: p=.0001 ***
LANGUAGE: p=.15
AGE*LANGUAGE (interaction): p=.03 *

From this alone, we’d note that age significantly affects /a/ production, language does not, but there is a significant interaction, so it’s possible that the age effect depends on language, or likewise that a language effect depends on age. The post-hoc tests on the significant age variable (since it has 3+ levels) and the significant interaction may look like the following:

Young vs. Middle: p=.8
Young vs. Old: p=.02 *
Middle vs. Old: p=.001 **
Young Spanish vs. Young Catalan: p=.01 *
Middle Spanish vs. Middle Catalan: p=.01 *
Old Spanish vs. Old Catalan: p=.79

The AGE post-hoc (on the left) reveals that the Young and Middle speakers do not produce /a/ differently from one another, though each produce /a/ differently than the Old speakers. The INTERACTION post-hoc (on the right) reveals that there actually is a language effect; it simply depends on age. For Old speakers, Spanish and Catalan /a/ are not different. For Young and Middle speakers, however, Spanish /a/ is not produced the same as Catalan /a/. A manual inspection of the language difference for Young speakers vs. Middle speakers (i.e., plot the results or envision a graph of the hertz figures above) shows a unique direction of effect: for Young speakers, Catalan /a/ has a lower F1 than Spanish /a/, but for Middle speakers, Catalan /a/ has a higher F1 than Spanish /a/.

Thus, ANOVA outputs are great for showing which independent variables are significant or not, but they don’t explicitly show you different directions of effect. To get these, you need to either plot/visualize the results or use the tapply command in R.
A regression output differs from an ANOVA in its presentation (i.e., the actual math/results are IDENTICAL). It contains an independent variable intercept/reference level, which is a combination of the alphabetically-/numerically-first level of every independent variable. The p-value for the intercept (row1 below) indicates whether or not the dependent variable (F1 of /a/) is significantly different from a value of 0. The remaining p-values indicate whether or not the level in question is significantly different from the intercept. The results below could be obtained from a regression on the same dataset:

<table>
<thead>
<tr>
<th>Estimate/β-coefficient</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row1: INTERCEPT (Middle Catalan):</td>
<td>+800</td>
</tr>
<tr>
<td>Row2: Spanish:</td>
<td>-400</td>
</tr>
<tr>
<td>Row3: Young:</td>
<td>-400</td>
</tr>
<tr>
<td>Row4: Old:</td>
<td>-400</td>
</tr>
<tr>
<td>Row5: Young-Spanish</td>
<td>+800</td>
</tr>
<tr>
<td>Row6: Old-Spanish</td>
<td>+400</td>
</tr>
</tbody>
</table>

We’ll interpret row by row, noting again that this result is IDENTICAL to that of the ANOVA, just presented in a unique format.

**Row1:** The significant intercept means that the F1 of /a/ for Middle-aged Catalan speakers is significantly different from 0 hertz, specifically 800 hertz greater than 0 (hence the positive coefficient), or 800 hertz.

**Row2:** From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Middle Spanish speakers (i.e., only language is shifting, with age remaining unchanged from the intercept). The significant p-value suggests that the F1 for Middle Spanish speakers is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

**Row3:** From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Young Catalan speakers (i.e., only age is shifting, with language remaining unchanged from the intercept). The significant p-value suggests that the F1 for Young Catalan speakers is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

**Row4:** From the intercept (Middle Catalan) value of 800 hertz, we now look to a possible significant difference in hertz for Old Catalan speakers (i.e., only age is shifting, with language remaining unchanged from the intercept). The significant p-value suggests that the F1 for Old Catalan speakers group is 400 hertz lower than the intercept, i.e., 800 – 400 = 400 hertz.

**Row5:** Calculating the F1 production for Young Spanish speakers, we start from the intercept of 800 and subtract 400 (as row 2 is significant) and then subtract another 400 (as row 3 is significant), yielding 800-400-400 = 0 hertz. At this point, i.e., for Young Spanish speakers, because row5 is significant, we now add 800 (effectively nullifying the prior main effects of age and language) to come to a predicted hertz production for Young Spanish speakers of 800 hertz (or 800-400-400+800).

**Row6:** Calculating the F1 production for Old Spanish speakers, we start from the intercept of 800 and subtract 400 (as row 2 is significant) and then subtract another 400 (as row 4 is significant), yielding 800-400-400 = 0 hertz. At this point, i.e., for Old Spanish speakers, because row6 is significant, we now add 400 (effectively nullifying the prior main effect of language) to come to a predicted hertz production for Old Spanish speakers of 400 hertz (or 800-400-400+400).

Accordingly, regression outputs are great for noting directions of effect, but require some mental math in order to fully interpret and additionally aren’t as transparent as ANOVAs for individual independent variables’ statistical significance.
Reporting a Linear Regression ('ContinuousDV_ANOVA_LinearReg' sheet in Example Datasets File):

<table>
<thead>
<tr>
<th></th>
<th>( \beta ) coefficient (in DV units)</th>
<th>Standard Error</th>
<th>Test Statistic (t, F, z, etc.)</th>
<th>p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept/Reference Level (Monolingual, Age 18-30)</td>
<td>53.50</td>
<td>2.06</td>
<td>25.95</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Early Bilingual</td>
<td>27.06</td>
<td>2.83</td>
<td>9.55</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Late Bilingual</td>
<td>26.83</td>
<td>2.83</td>
<td>9.47</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Age 50-60</td>
<td>-20.50</td>
<td>3.02</td>
<td>-6.79</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>Early:50-60 (interaction)</td>
<td>3.94</td>
<td>4.31</td>
<td>0.92</td>
<td>0.362</td>
</tr>
<tr>
<td>Late:50-60 (interaction)</td>
<td>3.00</td>
<td>4.31</td>
<td>-0.70</td>
<td>0.487</td>
</tr>
</tbody>
</table>

Report as this ↑

Report as the table above! ↓

R gives you this; Report as the table above!

Prose uses this format ➤️

Significant effect of Profile: \( \chi^2(2,129) = 118.241 ; p<.0001 \)

Significant effect of Age: \( \chi^2(1,129) = 29.014 ; p<.0001 \)

No significant effect interaction: \( \chi^2(2,129) = 0.919 ; p=.6317 \)
### Reporting a Logistic Regression (`BinaryDV_LogisticReg` sheet in Example Datasets File):

<table>
<thead>
<tr>
<th>Intercept/Reference Level (Saks, Emphatic, floorR)</th>
<th>β coefficient (in log-odds / logits)</th>
<th>Standard Error</th>
<th>Test Statistic (t, F, z, etc.)</th>
<th>p value</th>
<th>% Rhotic Loss (a)</th>
<th>Factor Weight (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-1.272</td>
<td>0.931</td>
<td>-1.366</td>
<td>0.172</td>
<td>[50%]</td>
<td>[0.50]</td>
</tr>
<tr>
<td>Kleins</td>
<td>2.241</td>
<td>0.199</td>
<td>11.24</td>
<td>&lt;.0001</td>
<td>90.39%</td>
<td>.90</td>
</tr>
<tr>
<td>Macys</td>
<td>0.441</td>
<td>0.138</td>
<td>3.197</td>
<td>.0014</td>
<td>60.85%</td>
<td>.61</td>
</tr>
<tr>
<td>Normal</td>
<td>0.331</td>
<td>0.127</td>
<td>2.616</td>
<td>.0089</td>
<td>58.20%</td>
<td>.58</td>
</tr>
<tr>
<td>fouRth</td>
<td>1.367</td>
<td>0.532</td>
<td>2.568</td>
<td>.0102</td>
<td>79.69%</td>
<td>.80</td>
</tr>
<tr>
<td>Speech Rate</td>
<td>0.004</td>
<td>0.005</td>
<td>0.706</td>
<td>0.4805</td>
<td>[0.1%]</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Calculations of % Rhotic Loss and Factor Weight (the latter applies only to nominal variables, hence “N/A” for speech rate)**

*All β-coefficients lacking statistical significance are to be treated as 0 in calculations!*

(a)/(c) For the **intercept row alone**, the calculation of % PREDICTED rhotic loss (i.e., these aren’t simply %s calculated from your raw data) uses the following formula: $\exp(\beta)/(1 + \exp(\beta)) = $ # # (adjust the decimal two spaces to the right to yield the %)

Accordingly, we’d have the following for the intercept row: $\exp(0)/(1 + \exp(0)) = .5$ ; 50%

(a) For all other (non-intercept) rows, the calculation of % PREDICTED rhotic loss uses the following formula:

\[
\exp(\text{Intercept} + \text{Non-Intercept})/(1 + \exp(\text{Intercept} + \text{Non-Intercept})) = # #
\]

(adjust the decimal by 2 for a %)

Accordingly, we’d have the following calculations of predicted rhotic % for each of the remaining nominal IV levels:

- Kleins: $\exp(0 + 2.241)/(1 + \exp(0 + 2.241)) = 90.39%
- Macys: $\exp(0 + 0.441)/(1 + \exp(0 + 0.441)) = 60.85%
- Normal: $\exp(0 + 0.331)/(1 + \exp(0 + 0.331)) = 58.20%
- fouRth: $\exp(0 + 1.367)/(1 + \exp(0 + 1.367)) = 79.69%

(b) Since this IV is not significant, rhotic loss would change by 0% for every unit increase in speech rate, i.e., speech rate has no effect. Still, if we pretend there is statistical significance, the calculation of % change in rhotic loss for every unit increase in speech rate is simply the difference between 50% (a log-odds of 0) and the % corresponding to the log-odds found here, namely $0.004 \Rightarrow \exp(0.004)/(1 + \exp(0.004)) = .501; or 50.1% \Rightarrow 50.1% - 50% = 0.1%$

For every 1-unit increase of speech rate, the model would predict an increase in rhotic loss of 0.1%... had the IV been significant.

(c) The calculation of factor weight (nominal rows) uses the following formula: $\exp(\beta)/(1 + \exp(\beta)) = # #$ (leave the decimal; round to hundredths). Thus, factor weights are as follows: Intercept: $\exp(0)/(1 + \exp(0)) = .50$; Kleins: $\exp(2.241)/(1 + \exp(2.241)) = .90$; Macys: $\exp(0.441)/(1 + \exp(0.441)) = .61$; etc...

**NOTE:** Only when the intercept is non-significant (thus Intercept: $\beta=0$) will DV %s be equivalents of factor weights for non-intercept nominal rows.
Between-Subjects ANOVA (Results appear as individual IVs that either are or are not significant)

(All IVs are nominal and between-subjects [each participant can only contribute for 1 level], DV = continuous, no random effects)

LEGEND: Italics - Store upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

DV ; IV ; V : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: IV1 + IV2 + IV3 + IV4 (etc...)

IVdump - Interacts between IVs are denoted with an asterisk instead of a plus sign: IV1 + IV2 * IV3 + IV2 * IV4 (etc...)

1a) install.packages("car")  
1b) install.packages("pivottabler")

2a) library(car)  
2b) library(pivottabler)

3a) DataName <- read.csv(file.choose(),header=T)  
3b) head(DataName)  
3c) summary(DataName)  
3d) qhpvt(DataName, "IV1", c("IV2", "IV3"), "n")

4) attach(DataName)  
[No longer need to refer to a V as DataName$V]

5) leveneTest(DV ~ IV1 * IV2 * IV3, data=DataName)  
[Equal between-subjects residuals variance check (aka heteroscedasticity); you want p > 0.05, otherwise you may need to transform DV or use a non-parametric test.]

6) ModelName = aov(DV ~ IVdump, data=DataName)

7a) Residuals = residuals(ModelName)  
7b) shapiro.test(Residuals)

8a) summary(ModelName)  
8b) tapply(DV, IV, mean)

9a) TukeyHSD(ModelName, "IV1")  
9b) TukeyHSD(ModelName, "IV1:IV2")

10a) r2 = lm(formula = DV ~ IVdump, data=DataName)  
10b) summary(r2)

11) boxplot(DV ~ IV1 + IV2 + IV3)  
[Quick plot to visualize results]
Within-Subjects (aka Repeated Measures or Mixed Model) ANOVA (Results appear as individual IVs that are/aren’t significant) (All IVs are nominal and at least 1 is within-subjects [all participants contribute for all levels], DV = continuous, no random effects)

Legend:

*Italic* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R. 

*Words* - Create your own label, colors show continuity of label

**DV ; IV ; V** : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

**IVdump** - Name of the IVs from Excel you want to consider, separated by plus signs: IV1 + IV2 + IV3 + IV4 (etc...)

**IVdump** - Interactions between IVs are denoted with an asterisk instead of a plus sign: IV1 + IV2 * IV3 + IV2 * IV4 (etc...)

1a) install.packages("car")

1b) install.packages("emmeans")

1c) install.packages("ez")

1d) install.packages("pivottabler")

2a) library(car)

2b) library(emmeans)

2c) library(ez)

2d) library(pivottabler)

3a) DataName <- read.csv(file.choose(),header=T) [Find your .csv file! This file must consist of a single sheet only!]

3b) head(DataName) [Reminds you of V names; confirmation of uploading the right file]

3c) summary(DataName) [Confirms that ONLY IVs of a #/Quantity have median/mean/max/min stats computed]

3d) qhpvt(DataName, "IV1", c("IV2", "IV3"), "n()") [Creates table for nominal IVs to assess cell counts; permits assessment of data as 100% balanced or not for step 5a)]

4) attach(DataName) [No longer need to refer to a V as DataName$V]

5a) Sphericity = ezANOVA(DataName, dv = DV, wid = SubjectNumber, within = .(IV1, IV2), between = .(IV3, IV4), type = #) [Mauchly Sphericity Test, i.e., equal residual variances, for any within-subjects IVs with 3+ levels. IVs in the “within” parentheses are within-subjects, IVs in the “between” parentheses are between-subjects, and type is either 2 for 100% balanced cell counts, or 3 for not 100% balanced cell counts (see step 3d). For definitions of within-vs. between-subjects, see green text at top of this page and the prior page]
5b) **Sphericity**

[Ignore everything and just look at Sphericity test; you want \( p > 0.05 \), otherwise you may need to transform DV or use a non-parametric test. **Remember that this only applies to within-subjects IVs with 3+ levels!** If yours only has/have 2 levels, then you should skip step 5 entirely!]

6) `leveneTest(DV ~ IV1 * IV2 * IV3, data=DataName)`

[Equal between-subjects residuals variance check (aka heteroscedasticity); you want \( p > 0.05 \), otherwise you may need to transform DV or use a non-parametric test.]

7) `ModelName = aov(DV ~ IVdump + Error(SubjectNumber/(IV1 * IV2)), data=DataName)`

[IVs after `SubjectNumber` are all within-subjects IVs]

8a) `Residuals = residuals(ModelName)`

8b) `shapiro.test(Residuals)`

[Residuals normality check; you want \( p > 0.05 \), otherwise you may need to transform DV or use a non-parametric test.]

9a) `summary(ModelName)`

9b) `tapply(DV, IV, mean)`

[Results in ANOVA format (p value for each IV); for direction of effect, see 9b]

9c) `summary(YetAnotherName)`

9d) `summary(OneLastName)`

[Computes DV per IV level; Run this on each significant IV to interpret its effect direction]

10a) `NewName = emmeans(ModelName, ~ IV)`

10b) `pairs(NewName)`

[Post-hoc for significant nominal IV with 3+ levels]

11a) `YetAnotherName = emmeans(ModelName, pairwise ~ IV1 | IV2)`

11b) `OneLastName = emmeans(ModelName, pairwise ~ IV2 | IV1)`

11c) `summary(YetAnotherName)`

11d) `summary(OneLastName)`

[Post-hoc for significant interaction between 2 nominal IVs; note that you basically run things twice, flipping order of IVs around]

12a) \( \hat{r}^2 = \text{lm(formula = DV ~ IVdump, data=DataName)} \)

12b) `summary(`\( \hat{r}^2 `)`)

[Obtains the \( r^2 \) for the model; ignore the output and just search for the *adjusted* \( r^2 \) value]

\[ \text{higher } r^2 = \text{more DV variance accounted for by model; } .82 = 82\% \]

13) `boxplot(DV ~ IV1 + IV2 + IV3)`

[Quick plot to visualize results]
Fixed Effects Linear Regression (Results appear as coefficients for all-1 levels per IV)

(No restriction on IVs, DV = continuous, no random effects)

Reference/Intercept (for the IVs) = Alphabetical, See Step 5 to change!

Legend:

*Italicics* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

**DV ; IV ; V :** Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV)

**IVdump** - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1 + IV2 + IV3 + IV4** (etc...)

**IVdump** - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1 + IV2 * IV3 + IV2 * IV4** (etc...)

1a) `install.packages("car")`

1b) `install.packages("emmeans")`

1c) `install.packages("lmtest")`

1d) `install.packages("pivottabler")`

1e) `install.packages("ggplot2")`

2a) `library(car)`

2b) `library(emmeans)`

2c) `library(lmtest)`

2d) `library(pivottabler)`

2e) `library(ggplot2)`

3a) `DataName <- read.csv(file.choose(),header=T)`

3b) `head(DataName)`

3c) `summary(DataName)`

3d) `qhpvt(DataName, "IV1", c("IV2", "IV3"), "n()")`

4) `attach(DataName)`

5) `DataName = within(DataName, IV <- relevel(IV, ref = "NewIVReferenceLevel"))`

[OPTIONAL: For REGRESSION OUTPUT: Set a not-alphabetically-first level of any one nominal IV to be the reference/intercept in the model (step 9a). Run this code multiple times to change the reference / intercept for various nominal IVs, one at a time]

[EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Male"` Changes intercept from Female to Male]

[EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Female"` Does nothing – Intercept was already Female]
6) **ModelName** = glm(DV ~ IVdump, data=DataName)

7a) Residuals=residuals(ModelName) [Residuals normality check; you want p > 0.05, otherwise you may need to transform DV or use a non-parametric test.]

7b) shapiro.test(Residuals)

8) bptest(ModelName) [Equal between-subjects residuals variance check (aka heteroscedasticity; you want p > 0.05, otherwise you may need to transform DV or use a non-parametric test.]

9a) summary(ModelName) [Results as a linear regression with $\beta$ coefficients and Model AIC]

9b) Anova(ModelName, type=2) [Results as an ANOVA with p values for each IV – type is either 2 for 100% balanced cell counts, or 3 for not 100% balanced cell counts. Refer to step 3d to assess equal counts for all nominal IVs. For direction of effect, see 9c]

9c) tapply(DV, IV, mean) [Computes DV per IV level; Run this on each significant IV to interpret its effect direction]

10a) emmeans(ModelName, list(pairwise ~ IV), adjust="tukey") [Post-hoc for significant Nominal IV with 3+ levels]

10b) emmeans(ModelName, list(pairwise ~ IV1*IV2*IV3), adjust="tukey") [Post-hoc for significant interaction between 2+ Nominal IVs]

11a) r2 = lm(formula = DV ~ IVdump, data=DataName) [Obtains the $r^2$ for the model; ignore the output and just search for the adjusted $r^2$ value]

11b) summary(r2) [higher $r^2$ = more DV variance accounted for by model, i.e., .82 = 82%]

12a) ggplot(DataName, aes(IV1, DV)) + geom_point(size=2, aes(color=IV2, shape=IV3)) [Quick plot to visualize results where IV1 is continuous and IV2 and 3 are nominal]

12b) boxplot(DV ~ IV1 + IV2 + IV3) [Quick plot to visualize results where all IVs are nominal]
Fixed Effects Logistic Regression (Results appear as coefficients for all minus one levels per IV) 
(No restriction on IVs, DV = binary [2 levels] nominal/discrete, no random effects) 
Reference/Intercept for DV = REVERSE-Alphabetical, See Step 5 to change! 
Reference/Intercept for IVs = Alphabetical, See Step 5 to change! 

Legend: 
Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R. 
Words - Create your own label, colors show continuity of label 
DV ; IV ; V : Name of your Excel DV column; Name of your Excel IV column; Name of an Excel Column (either IV or DV) 
IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: IV1 + IV2 + IV3 + IV4 (etc...) 
Rowdump - From output of step 7a, the intercept is row 1, followed by row 2 (a level), row 3 (a level), etc. For an IV with only 2 levels, simply type the row # of the non-intercept level, such as: 5 (meaning this factor has an intercept level and the other level is in row 5 of the summary [step 7a] output). For an IV with 3+ levels, type the row # of the first non-intercept level, followed by a colon (:), then the row # of last level of that factor, such as: 2:5 (meaning this factor has an intercept level and the remaining 3 levels occupy rows 2, 3, 4, and 5 of the summary [step 7a] output).

1a) install.packages("car") 
[if there’s a pop-question about binary versions available and installing sources from a package needing compilation, say no ]
1b) install.packages("emmeans")
1c) install.packages("pscl")
1d) install.packages("pivottabler")
1e) install.packages("aod")  
[only if you want to force an ANOVA-like output]

2a) library(car)
2b) library(emmeans)
2c) library(pscl)
2d) library(pivottabler)
2e) library(aod)
[only if you want to force an ANOVA-like output]

3a) DataName <- read.csv(file.choose(),header=T) [Find your .csv file! This file must consist of a single sheet only!]
3b) head(DataName) [Reminds you of V names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY IVs of a #:Quantity have median/mean/max/min stats computed]
3d) qhpvt(DataName, "IV1", c("IV2", "IV3"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a V as DataName$V]
Justindavidson@berkeley.edu

5) `DataName` = within(`DataName`, `V` <- relevel(`V`, ref = "NewIVReferenceLevel/OLDDVReferenceLevel"))  
   [OPTIONAL: set a not-alphabetically-first level of any IV or the alphabetically-first level of the DV to be the reference/intercept in the model (step 7a). Run this code multiple times to change the reference/intercept for multiple Vs, one at a time. NOTE THAT A DV REFERENCE CHANGE IS COUNTERINTUITIVE: YOU WRITE THE DEFAULT/OLD LEVEL TO CHANGE IT]

   [EXAMPLE for an IV with "Male" and "Female" as levels: `ref` = "Male" Changes intercept from Female to Male]
   [EXAMPLE for an IV with "Male" and "Female" as levels: `ref` = "Female" Does nothing – Intercept was already Female]
   [EXAMPLE for a DV with "Full" and "Aspirated" as levels: `ref` = "Full" Changes intercept from Full to Aspirated]
   [EXAMPLE for a DV with "Full" and "Aspirated" as levels: `ref` = "Aspirated" Does nothing – Intercept was already Full]

6) `ModelName` = glm(`DV ~ IVdump`, data=`DataName`, family = "binomial")

7a) `summary(ModelName)`  
   [Results as a regression with log-odds coefficients (β) and model AIC]
   [Should you want to convert log-odds coefficients into decimal %s / factor weights (e.g. .71 = 71%), copy/paste the following into R, replacing X with the β coefficient: `exp(X)/(1 + exp(X))` OR, use the code `inv.logit(X)` after running steps 1 & 2 on the package gtools]
   [Lower AIC = better model fit]

   [The ANOVA-like output of individual IVs as significant or not significant is available for logistic regression, but cumbersome and not often reported. It is common to simply intuit which IVs are significant based on which IVs have levels with significant β coefficients or which interactions are significant.]

7b) `wald.test(b = coef(`ModelName`), Sigma = vcov(`ModelName`), Terms = `Rowdump`)`  
   [ANOVA-like output for an IV that, ignoring the intercept (in row 1), contains the level or levels specified by `Rowdump`]

8a) `emmeans(`ModelName`, list(pairwise ~ `IV`), adjust="tukey")`  
   [Post-hoc for significant Nominal IV with 3+ levels]

8b) `emmeans(`ModelName`, list(pairwise ~ `IV1*IV2*IV3`), adjust="tukey")`  
   [Post-hoc for significant interaction between 2+ Nominal IVs]

9) `pR2(`ModelName`)`  
   [Obtains the pseudo-$r^2$ for the model; the value you want is McFadden!]
   [higher $r^2$ = more DV variance accounted for by model; .82 = 82%]

10) `plot(table(`IV1`, `IV2`, `IV3`, `DV`), col=T, main="InsertTitleHere")`  
    [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on the + or – β coefficients from step 7a]
Justin Davidson  
justindavidson@berkeley.edu

**Mixed Effects Linear Regression** (Results appear as coefficients for all minus one levels per IV)
(No restriction on IVs, DV = continuous, Presence of 1+ random effects [at least 2 data points per Random IV level])

**Reference/Intercept (for the IVs) = Alphabetical, See Step 5 to change!**

Legend:

*Italics* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

*Words* - Create your own label, colors show continuity of label

**DV** or **IV; V; RIV**: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column

**IVdump** - Name of the IVs from Excel you want to consider, separated by plus signs: **IV1 + IV2 + IV3 + IV4** (etc...)

**IVdump** - Interactions between IVs are denoted with an asterisk instead of a plus sign: **IV1 + IV2 * IV3 + IV2 * IV4** (etc...)

**IVRandomdump** - For each random intercept, use the following notation with parentheses: (1|**RIV1**) + (1|**RIV2**) (etc...)

**IVRandomdump** - For a random slope for a specific random intercept, replace the “1” for the random intercept codes above with the **within-subjects** IV(s) of choice: **(IV1|RIV1) + (IV1|RIV2) + (IV2|RIV2)** (etc...)

---

**1a)** install.packages("afex")  
**1b)** install.packages("lmerTest")  
**1c)** install.packages("emmeans")  
**1d)** install.packages("r2glmm")  
**1e)** install.packages("pivottabler")  
**1f)** install.packages("ggplot2")  

[if there’s a pop-question about binary versions available and installing sources from a package needing compilation, say no ]

**2a)** library(afex)  
**2b)** library(lmerTest)  
**2c)** library(emmeans)  
**2d)** library(r2glmm)  
**2e)** library(pivottabler)  
**2f)** library(ggplot2)  

**3a)** DataName <- read.csv(file.choose(),header=T)  
**3b)** head(DataName)  
**3c)** summary(DataName)  
**3d)** qhpvt(DataName, "IV1", c("IV2", "IV3"), "n()")  

[Find your .csv file! This file must consist of a single sheet only!]

[Reminds you of V names; confirmation of uploading the right file]

[Confirms that ONLY IVs of a #/Quantity have median/mean/max/min stats computed]

[Creates table for nominal IVs to assess cell counts]

**4)** attach(DataName)  

[No longer need to refer to a V as DataName$V]
5) **DataName** = within(\texttt{DataName, IV} <- relevel(\texttt{IV, ref = "NewIVReferenceLevel"}))  

\textbf{[OPTIONAL: For REGRESSION OUTPUT:]} Set a not-alphabetically-first level of any one **nominal** IV to be the reference/intercept in the model (step 7a). Run this code multiple times to change the reference / intercept for various nominal IVs, one at a time

\begin{itemize}
  \item [EXAMPLE for an IV with "Male" and "Female" as levels: \texttt{ref = "Male"}] \textit{Changes intercept from Female to Male}
  \item [EXAMPLE for an IV with "Male" and "Female" as levels: \texttt{ref = "Female"}] \textit{Does nothing – Intercept was already Female}
\end{itemize}

6) **ModelName** = \texttt{lmer(DV ~ IVdump + IVRandomdump, data=\texttt{DataName, REML=FALSE})}  
\texttt{[An ‘observations cannot be < groups’ error means there is only 1 data point per RIV level. You can only use a fixed effects model!]}  

7a) \texttt{summary(ModelName)} \texttt{[results as a \textbf{regression} with $\beta$ coefficients and \textbf{model AIC}]}  
7b) \texttt{mixed(ModelName, DataName)} \texttt{[results as an \textbf{ANOVA} with $p$ values for each IV; For direction of effect, see 7c]}  
7c) \texttt{tapply(DV, IV, mean)} \texttt{[Computes DV per IV level; Run this on each significant IV to interpret its effect direction]}  

8a) \texttt{emmeans(ModelName, list(pairwise ~ IV), adjust="tukey")} \texttt{[Post-hoc for significant Nominal IV with 3+ levels]}  
8b) \texttt{emmeans(ModelName, list(pairwise ~ IV1*IV2*IV3), adjust="tukey")} \texttt{[Post-hoc for significant interaction between 2+ Nominal IVs]}  

9) \texttt{r2beta(ModelName, method ="nsj")} \texttt{[Obtains the $r^2$ for the model and for each individual IV]}  
\textit{[higher $r^2$ = more DV variance accounted for by model; .82 = 82%]}  

10a) \texttt{ggplot(DataName, aes(IV1, DV)) + geom_point(size=2, aes(color=IV2, shape=IV3))} \texttt{[Quick plot to visualize results where IV1 is continuous and IV2 and 3 are nominal]}  

10b) \texttt{boxplot(DV ~ IV1 + IV2 + IV3)} \texttt{[Quick plot to visualize results where all IVs are nominal]}
Mixed Effects Logistic Regression (Results appear as coefficients for all minus one levels per IV)
(No restriction on IVs, DV = binary & nominal/discrete, Presence of 1+ random effects [at least 2 data points per Random IV level])

Reference/Intercept for DV = REVERSE-Alphabetical. See Step 5 to change!
Reference/Intercept for IVs = Alphabetical, See Step 5 to change!

Legend:

*Italics* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words – Create your own label, colors show continuity of label

DV or IV; V; RIV: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: IV1 + IV2 + IV3 + IV4 (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: IV1 + IV2 * IV3 + IV2 * IV4 (etc...)

IVRandomdump - For each random intercept, use the following notation with parentheses: (1|RIV1) + (1|RIV2) (etc...)

IVRandomdump - For a random slope for a specific random intercept, replace the “1” for the random intercept codes above with the within-subjects IV(s) of choice: (IV1|RIV1) + (IV1|RIV2) + (IV2|RIV2) (etc...)

1a) install.packages("afex")
1b) install.packages("lmerTest")
1c) install.packages("emmeans")
1d) install.packages("r2glmm")
1e) install.packages("pivottabler")

[if there’s a pop-question about binary versions available and installing sources from a package needing compilation, say no ]

2a) library(afex)
2b) library(lmerTest)
2c) library(emmeans)
2d) library(r2glmm)
2e) library(pivottabler)

3a) DataName <- read.csv(file.choose(),header=T) [Find your .csv file! This file must consist of a single sheet only!]
3b) head(DataName) [Reminds you of V names; confirmation of uploading the right file]
3c) summary(DataName) [Confirms that ONLY IVs of a #/Quantity have median/mean/max/min stats computed]
3d) qhpvt(DataName, "IV1", c("IV2", "IV3"), "n()") [Creates table for nominal IVs to assess cell counts]

4) attach(DataName) [No longer need to refer to a V as DataName$V]
5) `DataName = within(DataName, V <- relevel(V, ref = "NewIVReferenceLevel/OLDDVReferenceLevel"))` [OPTIONAL: set a not-alphabetically-first level of any IV or the alphabetically-first level of the DV to be the reference/intercept in the model (step 7). Run this code multiple times to change the reference/intercept for multiple Vs, one at a time. **NOTE THAT A DV REFERENCE CHANGE IS COUNTERINTUITIVE: YOU WRITE THE DEFAULT/OLD LEVEL TO CHANGE IT**

   [EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Male"` Changes intercept from Female to Male]
   [EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Female"` Does nothing – Intercept was already Female]
   [EXAMPLE for a DV with "Full" and "Aspirated" as levels: `ref = "Full"` Changes intercept from Full to Aspirated]
   [EXAMPLE for a DV with "Full" and "Aspirated" as levels: `ref = "Aspirated"` Does nothing – Intercept was already Full]

6) `ModelName = glmer(DV ~ IVdump + IVRandomdump, data = DataName, family = "binomial")` [An ‘observations cannot be < groups’ error means there is only 1 data point per RIV level. You can only use a fixed effects model!]

7) `summary(ModelName)` [Results as a regression with log-odds coefficients (β) and model AIC]

   [Should you want to convert log-odds coefficients into decimal %s / factor weights (e.g. .71 = 71%), copy/paste the following into R, replacing X with the β coefficient: `exp(X)/(1 + exp(X))` OR, use the code `inv.logit(X)` after running steps 1 & 2 on the package `gtools`]

   [Lower AIC = better model fit]

8a) `emmeans(ModelName, list(pairwise ~ IV), adjust = "tukey")` [Post-hoc for significant Nominal IV with 3+ levels]

8b) `emmeans(ModelName, list(pairwise ~ IV1*IV2*IV3), adjust = "tukey")` [Post-hoc for significant interaction between 2+ Nominal IVs]

9) `r2beta(ModelName)` [Obtains the pseudo-$r^2$ for the model]

   [higher $r^2 = more DV variance accounted for by model; .82 = 82\%$]

10) `plot(table(IV1, IV2, IV3, DV), col=T, main = "InsertTitleHere")` [Quick mosaic plot to visualize results for nominal IVs where IV1 levels will be columns and IV2+ levels will be rows; Use this on an individual IV to confirm what you think the DV intercept is based on the + or – β coefficients from step 7]
Justin Davidson  

**Poisson Regression and Zero-Inflated Poisson Regression** (Results appear as coefficients for all minus one levels per IV)  
(No restriction on IVs, DV = count data with only 1 level [Your Excel has no DV column! It’s just a list of DV occurrences!])  

**Reference/Intercept for IVs = Alphabetical, See Step 5 to change!**

Legend:

*Italics* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.  
**Words** - Create your own label, colors show continuity of label  

**IV**: RIV: Name of Excel IV column; Name of Excel Random IV column  

**IVdump** - Name of the IVs from Excel you want to consider, separated by plus signs:  

IV dump: IV1 + IV2 + IV 3 + IV4 (etc..)  

**IVdump** - Interactions between IVs are denoted with an asterisk instead of a plus sign:  

IV dump: IV 1 + IV 2 * IV 3 + IV 2 * IV 4 (etc..)  

**IVRandomdump** - For each random intercept, use the following notation with parentheses:  

IV Random dump: (1|RIV1) + (1|RIV2) (etc..)  

**IVRandomdump** - For a random slope for a specific random intercept, replace the “1” for the random intercept codes above with the within-subjects IV(s) of choice:  

IV Random dump: (IV1|RIV1) + (IV1|RIV2) + (IV2|RIV2) (etc..)

---

1a) `install.packages("afex")`  
1b) `install.packages("lmerTest")`  
1c) `install.packages("emmeans")`  
1d) `install.packages("r2glmm")`  
1e) `install.packages("ggplot2")`  
1f) `install.packages("GLMMadaptive")`  
1g) `install.packages("pscl")`  

2a) library(afex)  
2b) library(lmerTest)  
2c) library(emmeans)  
2d) library(r2glmm)  
2e) library(ggplot2)  
2f) library(GLMMadaptive)  
2g) library(pscl)  

3a) `DataName <- read.csv(file.choose(),header=T)`  
3b) `head(DataName)`  

4) `attach(DataName)`  

---

21
5) `DataName = within(DataName, IV <- relevel(IV, ref = "NewIVReferenceLevel"))`

[EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Male"` Changes intercept from Female to Male]

[EXAMPLE for an IV with "Male" and "Female" as levels: `ref = "Female"` Does nothing – Intercept was already Female]

6a) `Distribution = xtabs(~ IV1 + IV2 + IV3, DataName)`
6b) `[first step toward converting your observations into Frequency Counts]
   [Allows you to see cell counts; “lots” of 0s suggests the use of a Zero-Inflated Poisson Regression]

7a) `FreqCounts = as.data.frame(Distribution)`
7b) `FreqCounts`
7c) `ggplot(FreqCounts, aes(Freq)) + geom_histogram()`
   [data now has a “Freq” column as the DV, per IV cell]
   [visualizes your counts: focus on the left-most (x=0) bar. If this is tallest, you may need a Zero-Inflated P. Regression]

8a) `ModelName = glm(Freq ~ IVdump, data=FreqCounts, family = "poisson")`
8b) `ModelName = glmer(Freq ~ IVdump + IVRandomdump, data=FreqCounts, family = "poisson")`
8c) `ModelName = zeroinfl(formula = Freq ~ IVdump, data=FreqCounts)`
8d) `ModelName = mixed_model(Freq ~ IVdump, random = ~ 1 | RIV, data=FreqCounts, family = zi.poisson(), zi_fixed = ~ IVdump, zi_random = ~ 1 | RIV)`

[Pseudo-r² for Poisson Regressions only]

9) `summary(ModelName)`
[Should you want to convert log-odds coefficients into DV counts, copy / paste the following into R, replacing X with the β coefficient: `exp(X)`]
[Lower AIC = better model fit]

[The ANOVA-like output of individual IVs as significant or not significant is available for logistic regression, but cumbersome and not often reported. It is common to simply intuit which IVs are significant based on which IVs have levels with significant β coefficients or which interactions are significant.]

10a) `emmeans(ModelName, list(pairwise ~ IV), adjust="tukey")`
10b) `emmeans(ModelName, list(pairwise ~ IV1*IV2*IV3), adjust="tukey")`

[Post-hoc for significant interaction between 2+ Nominal IVs]

11) `r2beta(ModelName)`
[obtains the pseudo-r² for Poisson Regressions only]
Model Comparison
(Evaluate which model is better – the one with/without an IV interaction? Random IV? Additional IVs? Poisson or Zero-Inflated Poisson? Etc...)

Legend:

Words - Create your own label, colors show continuity of label
DV or IV; V; RIV: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column

1) For all tests save Within-Subjects ANOVA (for which you should instead just run a mixed effects linear regression), proceed through ModelName = step. Make this your SIMPLER model (fewer IVs, RIVs, interactions, levels for specific IVs, etc.).

2) Hit UP arrow & rename ModelName to ModelNameComplex, adding complexity (more [R]IVs, interactions, zero-inflation, etc.) [Beyond adjusting the IVs & ModelName, be sure to check if the code before the DV needs modification, such as glm vs. lmer, etc.] [Repeat this step as many times as necessary, i.e., to compare 5 models, complete this step 4 times to create 4 additional models]

3a) vuong(ModelName, ModelNameComplex) [Use to compare Poisson Regression to 0-Inflated-P. Regression: ONLY Fixed FX!]
3b) anova(ModelNameComplex, ModelName, test="Chisq") [Use to compare a Fixed FX to a Mixed FX model: NO 0IP-Reg!]
3c) anova(ModelName, ModelNameComplex, test="Chisq") [Use when both models are either fixed or mixed FX: NO 0IP-Reg!]
   [Significant p value - use the more complex / Zero-Inflated model; Non-significant p value - use the simpler / Poisson model]
   [If comparing more than 2 models, expand the code by adding in the other model names, separated by commas, before test="Chisq"]

3d) For comparisons involving Zero-Inflated Poisson Regression Models, you must do things manually, via summary(ModelName):
   - Calculate the absolute value of the difference between degrees of freedom (DF) for each model
   - Calculate the absolute value of the difference between Log Likelihoods for each model and multiply by 2
   - pchisq(x2LogLik, df=DFdiff, lower.tail=FALSE)
   [Significant p value - use the more complex model; Non-significant p value - use the simpler model]
Step-Wise Regression (Results appear as the best-fit model, indicating which IVs significantly improve the model [vs. their absence])

Currently only available in R for FIXED effects Linear/Logistic/Poisson/0-Inflated Poisson Models, as well as MIXED Linear)

Legend:

Italics - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.

Words - Create your own label, colors show continuity of label

DV or IV; V; RIV: Name of Excel DV or IV column; Name of Excel Column (either IV or DV); Name of Excel Random IV column

IVdump - Name of the IVs from Excel you want to consider, separated by plus signs: IV1 + IV2 + IV3 + IV4 (etc...)

IVdump - Interactions between IVs are denoted with an asterisk instead of a plus sign: IV1 + IV2 * IV3 + IV2 * IV4 (etc...)

IVRandomdump - For each random intercept, use the following notation with parentheses: (1|RIV1) + (1|RIV2) (etc...)

IVRandomdump - For a random slope for a specific random intercept, replace the “1” for the random intercept codes above with the within-subjects IV(s) of choice: (IV1|RIV1) + (IV1|RIV2) + (IV2|RIV2) (etc...)

1) Select the appropriate test and complete all steps through ModelName = [Create as COMPLEX a model as possible!]

[Complex models have max number of IVs, max interactions, max random effects, etc.]

2) step(ModelName)

[runs the step-wise regression]

[The best-fit model appears at the bottom of the output after “Call:”, with interactions noted by IV1 : IV2 instead of IV1 * IV2]

[ IV1 * IV2 is the same as IV1 + IV2 + IV1:IV2 ]

3a) Hit the UP arrow until the ModelName code (via step 1 above) is displayed [sets up the overwriting of the complex model]

3b) Modify IVdump &/or IVRandomdump to match the (R)IVs displayed in best-fit model from step 2 [sets up best-fit model]

[Feel free to express interactions via * rather than : . See [note] under step 2, above]

4) Return to the original test steps and finish them as normal, starting with the summary(ModelName) step. [stats on best-fit model]
Chi-Square Test (Results appear as a single IV with DV proportions that are or are not significantly different across IV levels) (Nominal/Discrete IV [1 at a time!], DV = count data, no restriction on number of levels, No random effects)

Analysis: Is the distribution/proportion of DV consistent across IV levels (i.e., are all rows equal?)  

Legend:  

*Italic* - Stored upon quit, you only need to run these one first time, as well as once anytime you update / re-download R.  

*Words* - Create your own label, colors show continuity of label

1) `install.packages("rcompanion")`  
2) `library(rcompanion)`  
3) For DVs with 2+ levels, envision your data according to the following schematic:

<table>
<thead>
<tr>
<th>Level1 of IV</th>
<th>Counts of DV: Level1 (column1)</th>
<th>Level2</th>
<th>Counts of DV: Level2 (column2)</th>
<th>Level3</th>
<th>Counts of DV: Level3 (column3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level1</td>
<td>500 A</td>
<td>400 B</td>
<td>250 C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level2</td>
<td>550 D</td>
<td>455 E</td>
<td>305 F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level3</td>
<td>550 G</td>
<td>455 H</td>
<td>305 I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level4</td>
<td>550 J</td>
<td>455 K</td>
<td>305 L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3b) For DVs with only 1 level, envision your data according to the following schematic:

<table>
<thead>
<tr>
<th>Observations of DV:</th>
<th>Level 1 of IV: Level1 (column1)</th>
<th>Level 2 of IV: Level2 (column2)</th>
<th>Level 3 of IV: Level3 (column3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(row1)</td>
<td>500 A</td>
<td>400 B</td>
<td>250 C</td>
</tr>
<tr>
<td>Evenly Distributed DV</td>
<td>Sum of row1 values divided by # of columns</td>
<td>Sum of row1 values divided by # of columns</td>
<td>Sum of row1 values divided by # of columns</td>
</tr>
<tr>
<td>(row2)</td>
<td>D (1150/3)</td>
<td>E (1150/3)</td>
<td>F (1150/3)</td>
</tr>
</tbody>
</table>

4) `ModelName = matrix(c(A,B,C,D,E,F,G,H,I,J,K,L), nrow = MaxRow#, ncol= MaxColumn#, byrow= TRUE, dimnames = list(c("NameOf1stRow", "NameOf2ndRow", "NameOf3rdRow", "NameOf4thRow"), c("NameOfColumn1", "NameOfColumn2", "NameOfColumn3")))`  

[add or remove rows/columns names/values as necessary]

5) `ModelName`  

6a) `chisq.test(ModelName)`  
6b) `fisher.test(ModelName, workspace = 2e+9)`  

[Only use if all cells have at least 5 or more tokens]  
[Only use if any cell has 4 or fewer tokens; change 9 to different number if there is an error about workspace size]

7) `pairwiseNominalIndependence(ModelName, gtest = FALSE)`  

[Post-hoc to compare every pair of rows. Ignore all values without adjustment. For any pair that includes a cell with a token count of 4 or fewer, use p.adj.Fisher. For pairs where all cells have 5 or more tokens, use p.adj.Chisq.]
Unfortunate Problems

As commands are generally introduced to R via the installation and subsequent calling up of packages, the average R-user relies on independent researchers and other coding-inclined folks to constantly be creating and eventually updating them. Beyond the inevitable issue of package authors not keeping their packages up to date as new editions of R are released, another unfortunate problem is that not all packages perform tasks in parallel manners.

To give an example, consider the discrepancy in fetching the r2 for a fixed effects linear regression vs. a mixed effects one. Ideally, one might expect that a single “get r2”-like command might be used to retrieve this statistic for both tests. However, for the former test, the “summary” command won’t compute this statistic for a glm object (step [6]), so we must create a dummy “lm” object (steps [11a] and [11b]) simply to have the r2 presented to us. For the latter, an entirely novel package (r2glmm) is required (step [9] – and no, the r2beta command that the r2glm package offers does not work for fixed effects regressions – that would be too easy!). Overall, this is simply a case of inconvenience rather than a true problem (i.e., while having to juggle multiple creative strategies [codes] to get the same value for different tests is annoying, in the end r2 is still obtainable for each type of test).

True problems are cases where the available packages simply cannot give you what you need, leaving you stuck waiting for somebody to develop a package that does (or I suppose, you could always become a stats/coding expert and create your own package to address the issue!). Below is a list of gaps that I am aware of with this R tutorial, that is, cases where you currently (with this tutorial) CANNOT perform the listed tests. Some of these gaps may be cases where there currently is no package out there with said functionality; others (hopefully more likely) are cases where I simply am not aware that a package exists and could perform the task.

Should you be aware of any packages that allow you to perform any of the following, please let me know and I’ll check them out!

- Multinomial regression (for nominal DVs that have three or more levels) [I know packages are readily out there, I simply haven’t dabbled enough with them to select which set to showcase here. Suggestions welcome!]
- ANOVA-like output for a mixed effects logistic regression
- ANOVA-like output for any type of Poisson regression
- Mixed effects Zero-Inflated Poisson regression that allows for more than 1 random effect/slope to be included
- Model comparisons involving (mixed effects) Zero-Inflated Poisson regression [not a true gap, since I discuss how to pull the relevant test statistics and manually do some math to compare them... but ideally we should be just as able to use “vuong” or “anova” with these regression models as with all the rest, and currently we can’t.]
- Step-wise regression for mixed effects logistic models
- Step-wise regression for mixed effects Poisson models